# Program Control

Although execution of a procedure generally proceeds from top to bottom, procedure control statements enable you to channel a procedure's execution to different sections, to subroutines, or to the debugger. Program control statements can also halt execution of a procedure.

| Statement | Description |
| --- | --- |
| Call statement | Branches to an external subroutine. |
| Case statement | Provides a framework for conditional branching. Evaluates one or more test expressions, then executes the relevant code at the corresponding internal subroutine label. A Case statement must be introduced with Begin Case and terminated with End Case. |
| Debug statement | Invokes the debugger. |
| End statement | Terminates a Then or Else condition. |
| Function() | Similar to the "call @" syntax for subroutines, the Function syntax allows a function to be called whose name is indeterminate at compile time. |
| GoSub statement | Branches to an internal subroutine. |
| GoTo statement | Branches unconditionally to another portion of the procedure. |
| If condition expression | Same as single line If statement, except that the entire statement is evaluated as an expression, leaving only one of two values. |
| If...Then statement | Performs single-line or multiple-line logic. If not single-line, statements for both branches must be delimited with End. Multi-line conditional logic syntax is applicable to all conditional statements (such as Read or Locate statements). |
| Loop statement | Marks the beginning of a group of statements, where the group is to be executed repeatedly, until the test expression is satisfied. |
| Null statement | Used if a statement is called for, but no action is desired (example: If ... Then Null Else ...). |
| On... GoSub statement | Branches to local subroutines. When the subroutine is terminated, control returns to the statement following this one. |
| On...GoTo statement | Branches to one of several statement labels. |
| Return statement | Branches back to the statement following the calling statement. If used from a function, Return can pass a value back to the calling stored procedure or event handler. |